REMARKS

Claims 1-27, and 30-33 are pending in the present application. Claims 1-35 were presented for examination. Claims 28, 29, 34, and 35 have been cancelled by amendment.

In the office action mailed January 29, 2003 ("the Office Action"), the drawings were objected to as failing to comply with 37 C.F.R. 1.84(p)(5) and 37 C.F.R. 1.83(a). The specification was also objected to because of informalities. Claims 1-35 were rejected under 35 U.S.C. 112, first paragraph, as containing subject matter which was not described in the specification in such a way as to enable one skilled in the art to make or use the invention. Claims 1-25 were rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 6,339,559 to Bertin et al. ("the Bertin patent"). Claims 26-35 were rejected under 102(a) as being anticipated by U.S. Patent No. 6,172,935 ("the Wright patent").

With respect to the objection to the drawings, Figure 2 has been amended to show the logic circuitry 350. With respect to the objection under 37 C.F.R. 1.83(a), the drawings have been reviewed, and every feature of the invention specified in the claims is shown. However, if the Examiner maintains the objection to the drawings, Applicants request that the Examiner specifically point out which features are not shown in the drawings.

With respect to the objection to the disclosure, the specification has been amended to provide the serial number of the co-pending application referenced at the last line of page 6 and lines 1-2 of page 7.

With respect to the rejection of claims 1-35 under 36 U.S.C. 112, first paragraph, the specification and Figures 1-7 have been amended to overcome the rejection. Figures 1, 3, and 6 have been amended to include reference numbers for each block of the process, and the specification has been amended accordingly such that the blocks are described understandably in the specification. Figures 4, 5, and 7, and the specification, have been amended to describe the labels shown in the respective figures.

Figure 2 and the specification have been amended to describe the operation and interconnection of elements shown in Figure 2.

Applicants disagree with the Examiner's belief that all the steps as recited in method claims 2-25 are not described in the specification. Applicants direct the Examiner's attention to Figures 3 and 4, and to page 7, line 28-page 10, line 28, as well as to Figures 6 and 7,

and to page 11, line 6-page 13, line 10. The specification provides a description that is sufficient for one of ordinary skill in the art to practice the invention, as well as recognize that the Applicants were in possession of the invention at the time the application was filed.

No new matter has been added by the amendments to the Figures and the specification. If the Examiner maintains the rejection of claims 1-35 under 35 U.S.C. 112, first paragraph, Applicants would appreciate suggested amendments to the Figures and specification to overcome the rejection.

The disclosed embodiments of the invention will now be discussed in comparison to the prior art. Of course, the discussion of the disclosed embodiments, and the discussion of the differences between the disclosed embodiments and the prior art subject matter, do not define the scope or interpretation of any of the claims. Instead, such discussed differences merely help the Examiner appreciate important claim distinctions discussed thereafter.

Embodiments of the present invention include providing a method for parallel programming of a plurality of programmable elements, such as antifuses, in a plurality of devices under test (DUTs). As described in the present application, the method utilizes bank address latches and fuse address latches on each DUT to maintain the address of the antifuse of each DUT that is to be programmed. In this manner, once the bank and fuse addresses are latched by the respective latches, programming of that antifuse, for that particular device, can be initiated. Once the programming operation is initiated, another DUT can be set up for antifuse programming in the same manner, that is, bank and fuse addresses of an antifuse in a second DUT can be latched in the bank and fuse address latches of the second device to initiate antifuse programming therein. The antifuse in the second DUT can be different than the antifuse that is being programmed in the first DUT. Thus, different antifuses in different DUTs can be program in parallel.

The bank and fuse address latches allow for parallel antifuse programming in test systems that cannot provide different address signals to each DUT. Note that the address signals provide the bank and fuse addresses of the specific antifuse to be programmed. That is, each DUT will receive the same address signals even though the specific antifuse may not need to be programmed in every DUT. The parallel antifuse programming methods described in the present application program a first antifuse in a first DUT, and in parallel, program a second antifuse

(that is different than the first antifuse) in a second DUT, with the programming operation of the first and second antifuses overlapping for some period of time.

In contrast, the Bertin patent describes a system and method for programming antifuses in parallel having a reduced number of interconnects. Shown in Figure 2 of the Bertin patent is a conventional antifuse programming and reading circuit. As described in the Bertin patent, a programming transistor T1 is incorporated into the antifuse/fuse latch circuit that is selected by a decoding scheme when programming a unique antifuse. As further described in the Bertin patent, the conventional decoding scheme, which is shown in Figure 3, would require a one-to-one correspondence of signals to antifuses in order to uniquely program the antifuses. The invention proposed in the Bertin patent includes a predecoder that selects an antifuse bank by driving the N-band to the programming voltage. The T1 transistors of the antifuses of each antifuse bank is wired in parallel, with each T1 transistor wired to the corresponding T1 transistors in the other antifuse banks. In selecting antifuses for programming, the predecoder selects an antifuse bank by driving the associated N-band to the programming voltage, and then the T1 transistor of the specific antifuses of the plurality of antifuses of each bank are selected. Only those specific antifuses selected and located in the selected antifuse bank will be programmed The specific example described in the Bertin patent includes four antifuse banks, each bank having nine antifuses. In this particular arrangement, rather than needing 36 separate signals to activate the T1 transistors, only nine separate signals to activate the T1 transistors, and four N-band lines are necessary to select an individual antifuse or antifuses for programming.

As previously mentioned, claims 1-25 have been rejected under 35 U.S.C. 102(e) as being anticipated by the Bertin patent.

Claim 1 is patentably distinct from the teachings of the Bertin patent. Claim 1 recites a method for programming programmable elements of a plurality of memory devices, each memory device having at least a first and second programmable element thereon, the method comprising programming the first programmable element of a first memory device of the plurality, and programming the second programmable element of a second memory device of the plurality, the programming of the first and second programmable elements overlapping at least for a period of time. The Examiner has failed to show that the Bertin patent teaches the combination of elements recited by claim 1. As previously discussed, the Bertin patent teaches a

method of parallel programming antifuses in the same device, or if applied to multiple devices, parallel programming of the same antifuses in different devices. The Bertin patent, however, does not teach programming a *first* programmable element of a *first* memory device and programming a *second* programmable element in a *second* memory device where the programming of the first and second programmable elements *overlap* for at least a period of time.

For the foregoing reasons, claim 1 is patentably distinct from the Bertin patent. Therefore, the rejection of claim 1 under 35 U.S.C. 102(e) should be withdrawn.

Claim 8, 14, and 20 are also patentably distinct from the teachings of the Bertin patent. Claim 8 recites a method for programming programmable elements of a plurality of memory devices, comprising in a first memory device of the plurality, latching a first address corresponding to a programmable element located at a first location on the memory devices, initiating a programming event to program the programmable element in the first memory device located at the first location, in a second memory device of the plurality, latching a second address corresponding to a programmable element located at a second location on the memory devices, and before the completion of the programming event for the programmable element in the first memory device, initiating a programming event to program the programmable element in the second memory device located at the second location.

Claim 14 recites a method for an external tester to program programmable elements of a plurality of memory devices, comprising commanding a first of the memory devices to latch a first address corresponding to a programmable element located at a first location on the memory devices, commanding a second of the memory devices to latch a second address corresponding to a programmable element located at a second location on the memory devices, and programming the programmable elements of the first and second of the memory devices substantially concurrently.

Claim 20 recites a method for an external tester to program antifuses of a plurality of memory devices, comprising providing to a first memory device of the plurality an address corresponding to a programmable element to be programmed in the first memory device, providing a load command to the first memory device to latch the address, providing to a second memory device of the plurality an address corresponding to a programmable element to be

programmed in the second memory device, and providing a load command to the second memory device to latch the address.

The Bertin patent fails to anticipate claims 8, 14, and 20 because it does not teach the combination of limitations recited by the respective claims. As previously discussed with respect to claim 1, the Bertin patent does not describe programming different antifuses on different DUTs in parallel. Therefore, the rejection of claims 8, 14, and 20 under 35 U.S.C. 102(e) should be withdrawn.

Similarly, claims 2-7, which depend from claim 1, claims 9-13, which depend from claim 8, claims 15-19, which depend from claim 14, and claims 21-25, which depend from claim 20, are also patentably distinct from the Bertin patent based on their dependency from a respective allowable base claim. That is, each of the dependent claims further narrows the scope of the claim from which it depends, and consequently, if a claim is dependent from an allowable base claim, the dependent claim is also allowable. However, because each claim in an application represents a different invention, the rejection of an independent claim does not necessarily result in the rejection of claims depending therefrom. For the foregoing reasons, the rejection of claims 2-7, 9-13, 15-19, and 21-25 under 35 U.S.C. 102(e) should be withdrawn.

As previously mentioned, claims 26-35 have been rejection under 35 U.S.C. 102(a) as being anticipated by the Wright patent.

Claims 26 and 30 are patentably distinct from the teachings of the Wright patent.

Claim 26 recites a memory device having external address terminals and data terminals, and further having an array of memory with redundant memory to replace memory cells therein in accordance with programmed programmable elements, the memory device comprising an antifuse bank address latch coupled to the address terminals for latching an antifuse bank address applied to the address terminals corresponding to a bank of antifuses including a programmable element to be programmed by a programming event, an antifuse address latch coupled to the address terminals for latching an antifuse address applied to the address terminals corresponding to the programmable element to be programmed by the programming event, logic circuitry coupled to the address latch and the data terminals, the logic circuitry receiving antifuse bank and antifuse address load commands applied to the data terminals and providing control signals to the antifuse bank address latch and antifuse address

latch in response to receiving the antifuse bank and antifuse address load commands to cause the antifuse bank address latch and the antifuse address latch to latch the respective addresses applied to the external address terminals as the antifuse bank and antifuse address corresponding to the programmable element to be programmed, all respectively, and programming circuitry coupled to the antifuse bank address latch and antifuse address latch and at least a portion of the programmable elements, the programming circuitry performing the programming event to program the programmable element corresponding to the latched antifuse bank and antifuse addresses.

Claim 30 recites a memory device having external address terminals and an array of memory with redundant memory to replace memory cells therein in accordance with programmed antifuses, the memory device comprising a bank address latch for latching a bank address corresponding to an antifuse bank in which an antifuse to be programmed by a programming event is located, a fuse address latch for latching an antifuse address corresponding to the antifuse to be programmed by the programming event, and logic circuitry coupled to the bank and fuse address latches, the logic circuitry receiving address load commands and providing control signals to the bank and fuse address latches in response thereto to cause the latches to latch the respective addresses applied to the external address terminals as the address corresponding to the programmable element to be programmed.

Claims 26 and 30 are not anticipated because the Wright patent fails to teach the combination of limitations recited by claims 26 and 30. For example, the Wright patent fails to teach the antifuse bank address latch, the antifuse address latch, the logic circuitry, and the programming circuitry in the arrangement recited in claim 26. Additionally, the Wright patent fails to teach analogous circuitry performing the same operations as the antifuse bank address latch, the antifuse address latch, the logic circuitry, and the programming circuitry as recited in claim 26. As noted by the Examiner, the Wright patent merely teaches the conventional elements of a synchronous memory device, such as external command and address busses, control logic, a dual bank memory array including redundant memory, row and column address latches, and fuse set circuitry. The Wright patent fails to disclose the combination of elements recited by claim 26.

Similarly, the Wright patent does not teach the combination of elements recited by claim 30. The Wright patent fails to disclose the bank address latch, the fuse address latch, and the logic circuitry in the arrangement and performing the same operations as described in claim 30. None of the elements that the Examiner has indicated as being disclosed by the Wright patent anticipate the elements of claim 30, nor are they analogous with any of the elements recited by claim 30. For example, as described in the Wright patent, the row and column address latches latch memory addresses of memory cells to be accessed, and do not latch any bank or antifuse address information of antifuses to be programmed.

Claim 32 is not anticipated by the Wright patent as well. Claim 32 recites a computer system having a memory device that is similar to that recited in claim 26. The Wright patent fails to describe the combination of elements recited by claim 32. As previously discussed with respect to claim 26, the Wright patent fails to describe the antifuse bank address latch, the antifuse address latch, the logic circuitry, and the programming circuitry in the arrangement recited in claim 32, and further fails to teach analogous circuitry performing the same operations as the antifuse bank address latch, the antifuse address latch, the logic circuitry, and the programming circuitry as recited in claim 32.

For the foregoing reasons, claims 26, 30, and 32 are patentably distinct from the Wright patent. Therefore, the rejection of claims 26, 30, and 32 under 35 U.S.C. 102(a) should be withdrawn.

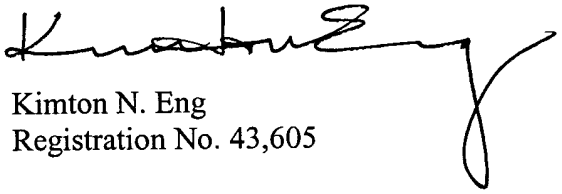
Claim 27, which depends from claim 26, claim 31, which depends from claim 30, and claim 33, which depends from claim 32 are also patentably distinct from the Wright patent based on their dependency from a respective allowable base claim. Therefore, the rejection of claims 27, 31, and 33 under 35 U.S.C. 102(a) should be withdrawn.

Attached hereto is a marked-up version of the changes made to the specification and claims by the current amendment. The attached page is captioned **"Version with Markings to Show Changes Made"**.

All of the claims remaining in the application are now clearly allowable. Favorable consideration and a Notice of Allowance are earnestly solicited.

Respectfully submitted,

DORSEY & WHITNEY LLP



Kimton N. Eng
Registration No. 43,605

KNE:asw

Enclosures:
Postcard
Fee Transmittal Sheet (+ copy)
Request for Drawing Change (Figures 1-7)
Transmittal for Filing Formal Drawings (8 Sheets, Figures 1-8)

1420 Fifth Avenue, Suite 3400
Seattle, WA 98101-4010
(206) 903-8800 (telephone)
(206) 903-8820 (fax)
h:\ip\documents\clients\micron technology\1000\501088.01\501088.01 amendment.doc

VERSION WITH MARKINGS TO SHOW CHANGES MADE

**In the Specification:**

Paragraph beginning at line 18 of page 1 has been amended as follows:

Figure 1 illustrates a flow diagram of a conventional method for programming antifuses of a memory device. A redundancy analyzer (RA) solution is calculated based on the location of any defective memory that is determined during testing at a step 102. Generally, the RA solution provides a solution for efficient use of the redundancy memory that is available on the memory device to replace the defective memory locations. The RA solution is then translated into bank and fuse addresses corresponding to the antifuses that should be programmed to remap defective memory locations to the redundancy memory that was calculated by the RA solution at a step 104. The bank and fuse addresses are stored programmatically in a fuse address array. Typically, the fuse address array is stored by the tester performing the testing. Conceptually, along one axis of the fuse address array is a list of the fuse identification (ID) of every possible antifuse on a memory device, and along a second axis is every memory device for a group of devices. The group of devices may be all of the memory devices of a single wafer, or all of the memory devices of a single lot of wafers. As a result of storing the data in the fuse address array, the particular antifuses that need to be programmed for each of the memory devices of the group is available for reference.

Paragraph beginning at line 27 of page 2 has been amended as follows:

In going through the process of programming the antifuses of the devices of each touchdown as shown in Figure 1 at steps 106-110, the tester sequences through the list of the fuse IDs of the fuse address array. For each fuse ID, the tester evaluates the stored information to determine if any of the devices in the touchdown need the antifuse corresponding to the current fuse ID to be programmed. If none of the devices require programming of the antifuse, then the tester continues to the next fuse ID. However, at a step 112, if any one of the devices in the touchdown need the antifuse corresponding to the current fuse ID to be programmed, then the SVM for each DUT having a device in need of antifuse programming is programmed with a data

value indicative of the bank address of the antifuse that needs to be programmed. For those devices in the touchdown that do not need the antifuse corresponding to the current fuse ID to be programmed, a null value is provided to the respective data pins instead.

Paragraph beginning at line 10 of page 3 has been amended as follows:

The fuse address of the antifuse to be programmed is commonly applied to the address terminals of each of the DUTs as in step 114. For those devices receiving a null data value, the fuse address has no effect. However, for the devices also receiving a data value indicative of a particular fuse bank, the application of the fuse address triggers a fuse blow event that programs the antifuse corresponding to the bank address applied to the data pins of the device and the fuse address applied to the address pins of the device as in step 116. The fuse address must be held valid throughout the entire duration of the fuse blow event, otherwise, the fuse blow event is terminated and the antifuse will not be programmed.

Paragraph beginning at line 18 of page 3 has been amended as follows:

Upon completion of the fuse blow event, at a step 118, the tester continues to the next fuse ID of the fuse address array, and the process of determining whether any of the devices in the touchdown require programming of the antifuse of the current fuse ID, and programming thereof if at least one device requires it, continues until all of the possible antifuses of the memory device are checked.

Paragraph beginning at line 1 of page 6 has been amended as follows:

Figure 2 illustrates a portion of a memory device 300 according to an embodiment of the present invention. The memory device 300 is similar to the conventional memory device[.], and such conventional elements have not been shown in order to avoid obscuring the present invention. For example, the memory device 300 includes a memory array (not shown) having [A memory array 320 includes] conventional redundant memory [that is] used to replace defective memory cells of the array. As with a conventional memory device, remapping of defective memory cells to the redundant memory is accomplished by programming a

programmable device, such as an antifuse. Conventional antifuse programming circuitry [(not shown)] 338 is used for programming of the antifuses. [As illustrated in Figure 2,] In the present embodiment, the memory array [320] is divided into four [fuse bank] array regions, [: Bank 0, Bank 1, Bank 2, and Bank 3. Within] each of [the] which has a fuse bank region[s] associated therewith. [are antifuses that can be uniquely identified by a bank address and a fuse address. Thus,] The four fuse bank regions 320a, 320b, 320c, and 320d are shown in Figure 2. Within each of the fuse bank regions are antifuses that can be programmed to replace defective memory cells of the associated memory array with the redundant memory. The antifuses can be uniquely identified by a bank address and a fuse address, and when programming of the antifuses takes place, the bank and fuse address are used to identify the antifuse that needs to be programmed. Programming of the antifuses in the four fuse bank regions 320a-320d is conventional and known to those ordinarily skilled in the art.

Paragraph beginning at line 13 of page 6 has been amended as follows:

Although the memory device 300 includes many aspects that are conventional, the memory device 300 is different in that it includes bank address latches 330, 332, and fuse address latches 340, 342, 344, 346, and logic circuitry 350 to enable parallel antifuse programming according to embodiments of the present invention. The bank address latches 330, 332, and the fuse address latches 340, 342, 344, and 346, are coupled to receive address signals A0-An from the external address terminals of the memory device 300. The logic circuitry 350 is coupled to external data terminals DQ0-DQ3[.], to receive command signals, and is further coupled to the bank address latches 330, 332 and the fuse address latches 340-346. As will be explained in more detail below, when[When] the memory device 300 is in an antifuse programming mode, command signals can applied to the memory device 300 via data terminals DQ0-DQ3 to instruct the loading of an address [that is] applied to the address terminals A0-An into either the bank address latches 330, 332, or the fuse address latches 340, 342, 344, 346. The logic circuitry 350 receives the command signals on DQ0-DQ3, and in response, generates BANKLAT0 and BANKLAT1 signals to instruct the bank latches 330, 332 to latch an antifuse bank address from the external address terminals, and generates signals CNTRL0-CNTRL3 to

instruct the fuse address latches 340-346 to latch an antifuse address from the external address terminals. The latched bank and antifuse addresses are then provided to the respective banks 320a-320d to uniquely select an antifuse for programming. As shown in Figure 2, conventional antifuse programming circuitry 338 is included for programming antifuses of the antifuse banks 320a-320d, as is well known.

Paragraph beginning at line 22 of page 6 has been amended as follows:

It will be appreciated that the memory device 300 includes additional conventional circuitry that has not been shown in Figure 2. These circuits are well known in the art, and consequently, will not be discussed herein for the sake of brevity. However, the description provided herein is sufficient to enable those of ordinary skill in the art to practice the invention. An alternative memory device suitable for use in embodiments of the present invention is described in more detail in co-pending U.S. Patent Application [No. 0X/XXX,XXX] No. 09/810,366 to Cowles, entitled CIRCUIT AND METHOD FOR TEST AND REPAIR and filed [March 5, 2001,] March 15, 2001, which is incorporated herein by reference.

Paragraph beginning at line 28 of page 7 has been amended as follows:

Figure 3 illustrates a flow diagram of an embodiment of the parallel antifuse programming process of the present invention. As with the conventional process, at a step 352, an RA solution is calculated based on the location of the defective memory cells and then at a step 354, translated into bank and fuse addresses corresponding to the antifuses that need to be programmed for each of the devices in a group. As previously mentioned, the group may be the devices on a single wafer, or the devices in a single lot of wafers. The bank and fuse address are stored in a data array as in the conventional process. However, in contrast with the conventional antifuse programming process, following the translation of the RA solution, the bank and fuse addresses of the antifuses for a first region of redundant memory of each device in a touchdown are loaded into a tester scramble memory SCRAM. In the present example that the antifuses for each device are programmed one fuse bank region at a time. However, it will be appreciated that the antifuse programming process can be performed for multiple regions at a time without departing from the scope of the present invention. Figure 5 illustrates a table 500 that provides

an example of the organization of the bank and fuse address data 502 and 504, respectively, stored in the tester SCRAM. Null values 506 are inserted into the bank and fuse addresses for the memory devices in the touchdown that do not require as many antifuses to be programmed. As will be explained in more detail below, this allows for the cycle of loading the bank and fuse addresses can be maintained for consistency during the fuse blow events. Although Figure 5 illustrates an example of a suitable arrangement for the bank and fuse addresses, as well as the appropriate load commands to apply to the DQs of the devices in the touchdown, it will be appreciated that other arrangements can be used as well without departing from the scope of the present invention.

Paragraph beginning at line 18 of page 8 has been amended as follows:

Following the loading of the bank and fuse addresses for the first fuse bank region into the tester SCRAM at a step 356, the process of having each memory device latch the respective bank and fuse address is carried out by programming the SVMs of each DUT in the touchdown to provide the appropriate load commands to the DQs of each of the memory devices. In the present example, loading of a respective bank and fuse address for each device of the touchdown is performed in sequence, with the sequence being repeated until all of the antifuses in the particular region for the worst case device are programmed as in steps 358-366. The antifuse programming process is then repeated for each of the remaining fuse bank regions until all of the antifuses of the memory devices have been programmed as in steps 368-372.

Paragraph beginning at line 1 of page 11 has been amended as follows:

An alternative embodiment of the present invention will be described with respect to [Figures 7 and 8,] Figures 6 and 7, and is directed to a tester having the capability of providing each device of a touchdown with unique address signals. This allows for the respective bank address and fuse address of each device to be loaded concurrently, thus, saving even more time than the previously described embodiment.

Paragraph beginning at line 6 of page 11 has been amended as follows:

Figure 6 illustrates a flow diagram of a parallel antifuse programming process according to an alternative embodiment of the present invention. An RA solution is calculated at a step 602 based on the location of the defective memory cells and then translated into bank and fuse addresses corresponding to the antifuses that need to be programmed for each of the devices in a group at a step 604. The bank and fuse address are stored in a data array as in the conventional process. The tester then applies the appropriate signals to enable the antifuse programming mode for all the devices in a touchdown.

Paragraph beginning at line 20 of page 11 has been amended as follows:

Following the entry into the antifuse programming mode, the bank and fuse addresses for the first antifuse in need of programming in each of the devices of the touchdown is loaded into the respective data registers for each of the DUTs at steps 606 and 608. As with the previous embodiment, the antifuses are programmed in a per fuse bank region basis. The timing diagram of Figure 7 illustrates programming the antifuses of fuse bank region 0. A 1H value is applied to the DQs of all of the memory devices in the touchdown to initiate a Load Bank Address Bank 0 and 1 command, and at a time T0, each of the memory devices in the touchdown latches in the bank address applied to the respective address terminals. The tester then switches the address signals to provide the respective fuse address to each of the memory devices and changes the data value applied to the DQs to 5H to initiate a Load Fuse Address Bank 0 command. At a time T1, the respective fuse addresses are latched by each memory device and a fuse blow event is triggered for each of the memory devices as shown in Figure 6 at step 610. While the fuse blow event is occurring, the bank and fuse addresses for next fuse in the region in need of programming for each memory device is programmed into the respective data registers. By programming the bank and fuse addresses for the next antifuse during the fuse blow event for the previous antifuse, the operation can essentially be hidden.

Paragraph beginning at line 18 of page 12 has been amended as follows:

The process of loading the bank and fuse addresses into the respective data registers for the next antifuse to be programmed in the region for each individual device, latching the bank address, then latching the fuse address and triggering a fuse blow event to program the antifuse corresponding to the respective bank and fuse addresses in each device, is repeated until all of the fuses in that region for the worst case device are programmed as shown in Figure 6 in steps 606-612. As with the previously described embodiment, null values are provided to the devices having less antifuses in need of programming than the worst case memory device and that are waiting for the programming of those antifuses to be completed.

Paragraph beginning at line 26 of page 12 has been amended as follows:

When programming of the antifuses for a region is completed, the process is repeated for the next region, and all remaining regions until all of the antifuses in need of programming for each device in the touchdown are programmed as shown in steps 606-614.

**In the Claims:**

Claims 28, 29, 34, and 35 have been cancelled.

Claims 26 and 32 have been amended as follows:

26. (Once amended) A memory device having external address terminals and data terminals, and further having an array of memory with redundant memory to replace memory cells therein in accordance with programmed programmable elements, the memory device comprising:

an antifuse bank address latch coupled to the address terminals for latching an antifuse bank address applied to the address terminals corresponding to a bank of antifuses including a programmable element to be programmed by a programming event;

an antifuse address latch coupled to the address terminals for latching an antifuse address applied to the address terminals corresponding to the [a] programmable element to be programmed by the [a] programming event;

logic circuitry coupled to the address latch and the data terminals, the logic circuitry receiving antifuse bank and antifuse address load commands applied to the data terminals and providing control signals to the antifuse bank address latch and antifuse address latch in response to receiving the antifuse bank and antifuse address load commands [thereto] to cause the antifuse bank address latch and the antifuse address latch to latch the respective addresses [an address] applied to the external address terminals as the antifuse bank and antifuse address corresponding to the [a] programmable element to be programmed, all respectively; and

programming circuitry coupled to the antifuse bank address latch and antifuse address latch and at least a portion of the programmable elements, the programming circuitry performing the [a] programming event to program the programmable element corresponding to the latched antifuse bank and antifuse addresses.

32. (Once amended) A computer system, comprising:

a data input device;

a data output device;

a processor coupled to the data input and output devices; and

a memory device coupled to the processor, the memory device comprising:

external address terminals;

data terminals;

an array of memory with redundant memory to replace memory cells therein in accordance with programmed programmable elements;

an antifuse bank address latch coupled to the address terminals for latching an antifuse bank address applied to the address terminals corresponding to a bank of antifuses including a programmable element to be programmed by a programming event;

an antifuse address latch coupled to the address terminals for latching an antifuse address applied to the address terminals corresponding to the [a] programmable element to be programmed by the [a] programming event;

logic circuitry coupled to the address latch and the data terminals, the logic circuitry receiving antifuse bank and antifuse address load commands applied to the data

terminals and providing control signals to the antifuse bank address latch and antifuse address latch in response to receiving the antifuse bank and antifuse address load commands [thereto] to cause the antifuse bank address latch and the antifuse address latch to latch the respective addresses [an address] applied to the external address terminals as the antifuse bank and antifuse address corresponding to the [a] programmable element to be programmed, all respectively; and

programming circuitry coupled to the antifuse bank address latch and antifuse address latch and at least a portion of the programmable elements, the programming circuitry performing the [a] programming event to program the programmable element corresponding to the latched antifuse bank and antifuse addresses.